GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Bachelor of Computer Applications

Level: UG

Course / Subject Code: BC04001041

Course / Subject Name: Programming with Python

w. e. f. Academic Year:	2025-26
Semester:	4
Category of the Course:	Core Course (CC)

Prerequisite:	Basic Concept of Programming
Rationale:	• To develop proficiency in creating applications using the Python Programming Language. • To understand and apply decision structures, looping and functions in programming. • To understand and apply lists, tuples, sets, dictionaries and strings in programming. • To understand and apply object oriented concepts in programming. • To understand and apply exceptions and file handling in programming.

Course Outcome:

After Completion of the Course, Student will able to:

No	Course Outcomes	RBT Level
01	Understand basic features and elements of python programming language.	AP
02	Demonstrate programs using decision structures, looping and functions.	AP
03	Demonstrate programs using lists, tuples, sets, dictionaries and strings.	AP
04	Demonstrate programs using object oriented concepts.	AP
05	Demonstrate programs using exceptions and file handling.	AP

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to Programming:** Basics of Programming language, Programming vs software, Python Programming language, Compiler vs Interpreter, Integrated Development Environment, Various resources. **Introduction to Python:** What is a program? Elements of Python language, Python block structure, Variables and assignment statements, Python objects, Data types, Operations, Type conversions, Input/output statements, Formatting, Built in functions.	5	10
2.	**Decision Structures:** Need, Forming conditions, if, if-else, if-elseif, nested if statements. **Looping statements:** Introduction, Built-in functions, Loop statements, Jump statements. **Functions:** Introduction, Types, Anatomy, Functions with parameters, void and value returning functions, Types of function arguments, Passing of object reference, Namespaces, Global variables, and scope.	9	20
3.	**Lists:** Introduction to Iterable object, List, Operations in List, Built-in functions and methods, Nested list, Mutability, Looping with list. **Tuples:** Introduction, Creation, Indexing and slicing, Operations, Immutability, Built-in functions and methods, Nested tuples, selection, looping with Tuple. **Sets and Dictionary:** Introduction, Creation, Operations, looping, Nesting. **Strings:** Introduction, Creation, Operations, Built-in methods and functions, Looping.	12	25
4.	**Object Oriented Programming:** Introduction, Classes and objects, Class attributes, Methods, Built-ins. **Inheritance:** Single, Multiple, Multi-level, Hybrid. **Advance class:** Overloading, Method overriding, Class method and static method, Abstract base class, Meta class.	10	25
5.	**Errors and exceptions:** Introduction, Handling exceptions, Multiple exceptions, Raising exceptions, Exceptions chaining, Built-in exceptions, User defined exceptions, Cleanup actions. **File handling:** Introduction to files, Types of files, Input output operations, File handling operations in text files, binary files and excel files.	9	20
	Total	**45**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
15	25	60	-	-	-

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

Text Book:

- Python Programming, S. Sridhar, J. Indumathi, V. M. Hariharan, Pearson Education

Reference Books:

- John V Guttag. "Introduction to Computation and Programming Using Python", 2nd Edition, Prentice Hall of India
- R Nageswara Rao, Core Python Programming, 2nd Edition, Dreamtech Press
- Python Programming, Sheetal Taneja, Naveen kumar, Pearson.

Open source software and website:

- Python 3.x, IDLE
- Faculty can suggest any online course from NPTEL, EdX, Coursera, Udemy , Agile and Scrum platforms (Based on availability of the course at the time of teaching learning as course availability remain changing.)

Suggested Practical List:

1	Write a program to generate first n prime numbers.
2	Write a program to find a maximum of three numbers using the conditional operator.
3	Write a program to find the second maximum of n numbers without using arrays.
4	Write a program to reverse the digits of a number using a while loop.
5	Write a program to convert numbers into words and print it.
6	Write a program using a for loop to determine the highest common factor or a greatest common divisor of a pair of a number.
7	Write a program to multiply two matrix.
8	Write a program using a function to compute average for variable length arguments of numbers.
9	Write a program using lamda function for finding the multiplication of two numbers.

10	Write a program using a function for finding whether the given number is an Armstrong number or not.
12	Write a program to reverse a list.
13	Write a program to add two lists index wise.
14	Write a program to iterate both lists simultaneously and display items from list1 in original order and items from list2 in reverse order.
15	Write a program to merge all the items in a list into one long string.
16	Write a program to create a tuple with different data types such as street address.
17	Write a program to check whether an element exists in tuple.
18	Write a program that swaps two tuples.
19	Write a program to reverse a tuple.
20	Write a program to copy selected elements from one tuple to another.
21	Given a pair of sets, generate a new set by collecting all the elements from one set those are absent from the other set.
22	Given a pair of sets, generate a new set containing all the elements from the two sets with no replicas.
23	Write a program to use discard() to remove the elements from a set.
24	Write a program to sort the given dictionary by its value.
25	Write a program to find the three highest values of matching dictionary keys.
26	Write a program that accepts a string and displays length of each word and length of string.
27	Write a program to elicit a string from the user and then determine the total number of characters included in a string.
28	Write a program to elicit a string from the user in lower or small case letters and then capitalize the initial characters of the string.
29	Write a program that obtains a user supplied string. Use the isDigits() to determine whether or not it contains digits.
30	Write a program to read, add, display and subtract two matrices using OOP.
31	Write a static method that check whether all words in the list starts with vowel.
32	Create a class for the bank which holds few details about the bank and define a destructor which

	should make all variables as “None” and print the message “The process Ends”.
33	Write a program that extends the class Employee. Derive a class Manager from Employee so that it lists all the details of manager as well as the details of employees working under the manager.
34	Write a program that extends the class Shape to calculate the volume of the cube and the cone.
35	Write a program to demonstrate single inheritance, multilevel inheritance and hierarchical inheritance.
36	Write a program to find the increment of all the elements in a list with value 2 by overloading the unary ‘+’ operator.
37	Write a program to develop user defined exception for 'Divide by Zero' error.
38	Write a program that accesses the elements of a list using range function and loop which handles index error.
39	Create a python script that handles multiple exceptions with separate blocks of code.
40	Write a program that reads the file line by line and count the number of lines.
41	Write a program that reads the first n lines of a given file.
42	Create a python script to count the number of words in a file.

Suggested Activities for Students, if any:

- ICT enabled Classroom teaching
- Case study
- Assignments
- Interactive classroom discussions

CO- PO Mapping:

Semester 4	**Course Name : Programming with Python**										
	POs										
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	-	-	-	-	-	1	-	-	-	-
CO2	3	-	2	-	1	-	1	-	-	-	-
CO3	3	1	3	-	1	-	1	-	-	-	-
CO4	3	2	3	-	1	-	1	-	-	-	-
CO5	3	2	3	-	1	-	1	-	-	-	-

* * * * * * *